COMP9321 Data Services Engineering

Semester 2, 2018

Week 1: Introduction

Data Services

- What? Data services are software services that encapsulate operations on key data entities relevant to the consumer
- Why? Data nowadays is stored in multiple systems and require multiple interfaces or mechanisms to interact with them. There are varying channels (e.g., legacy systems, Online, third-party) and mechanisms (e.g., event driven, on demand, batch process) that need to be served as well adding additional challenges to data services. Without an abstraction layer for data consumers that insulate them from this complexity we will end up with a spaghetti of point to point integrations between data sources and data consumers

Source: * https://www.infoq.com/articles/narayanan-soa-data-services

Let’s Go Deeper

Data Recording… The Beginning

Source: Wikipedia.org

Not that Deep

Data-oriented Services …

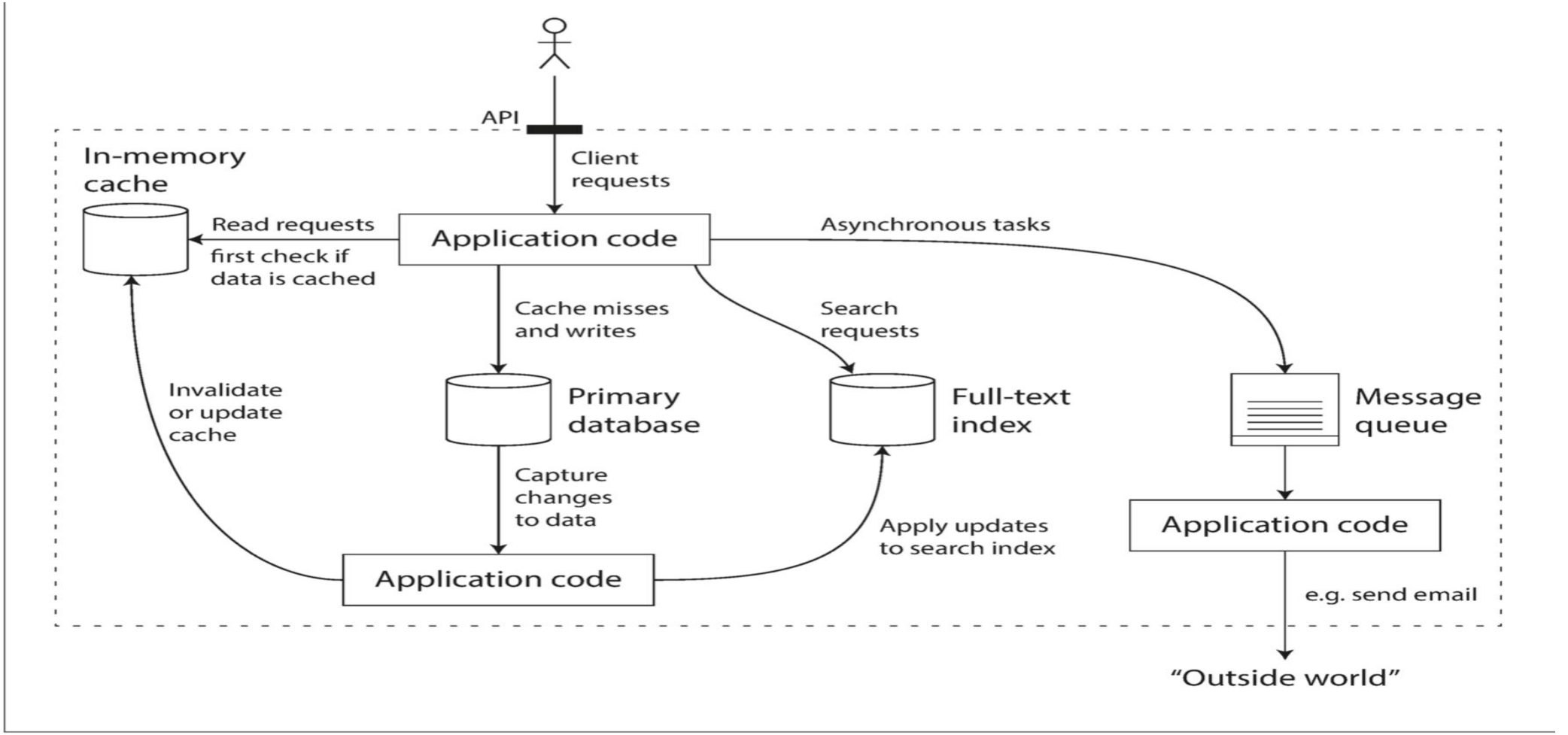


Figure 1-1. One possible architecture for a data system that combines several components.

Information Systems/Applications Integration

A set of services and solutions for bringing together disparate application and business processes as needed to meet the diverse information requirements of your customers, partners, suppliers and employees.

Motivations: Streamlining business operations, globalisation, competition, mergers and acquisition, new business models, technology development, etc.

- e.g., merger of two companies (data + processes)

Problems: systems to be integrated are not homogeneous.

- they are individually developed (ad-hoc) systems overtime
- some are “off-the-shelf” packages
- different execution platforms, technologies and business rules

Heterogeneity at different levels: language, platform, schema (data, process)

- Data integration, Process/Systems integration

Data Level Integration …

Data integration = combining data from different sources and providing users with a unified view over them

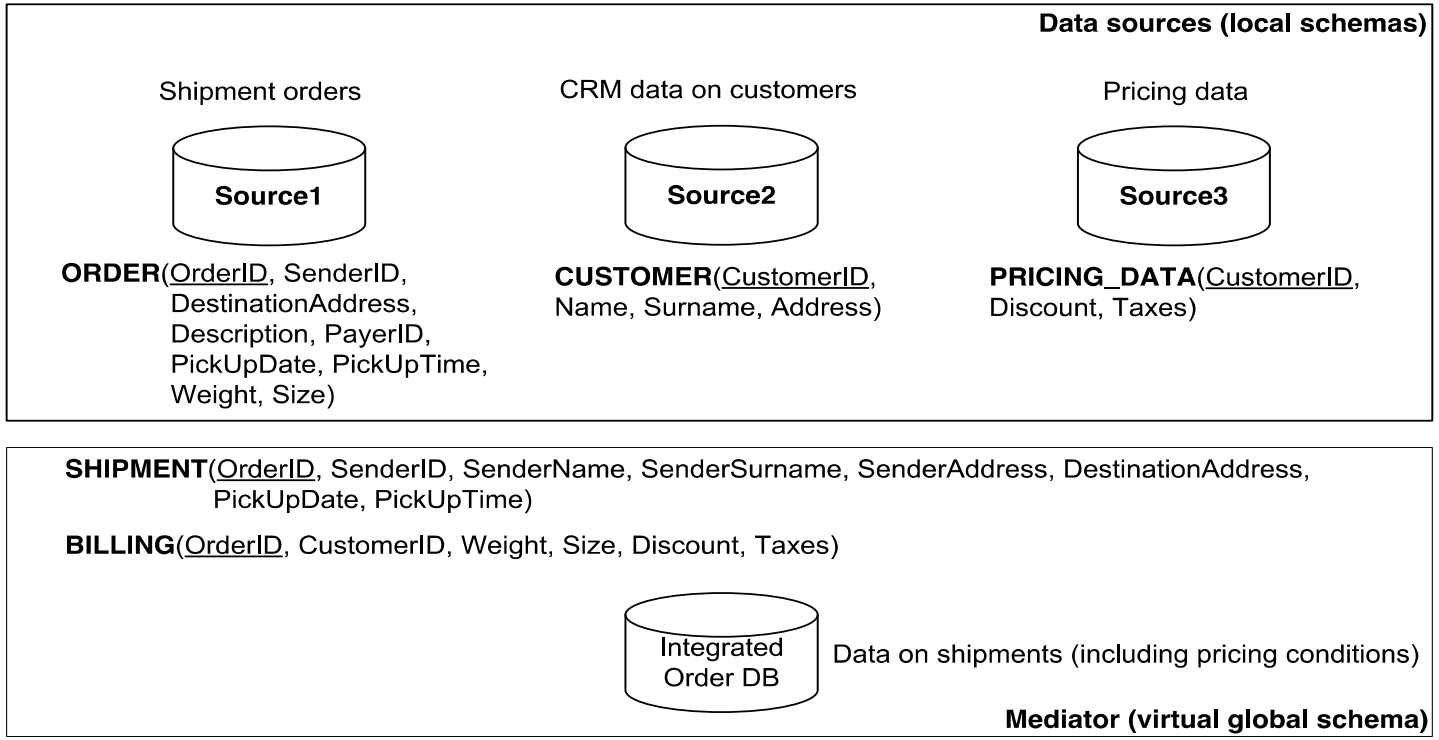


Fig. 2.2 Example of an integrated database storing shipment data extracted from different data sources. Each data source is characterized by a *local schema*. Data integration is performed according to a *vritual global schema* managed by the mediator.

Picture from Mashups: Concepts, Models and Architectures

Data Level Integration …

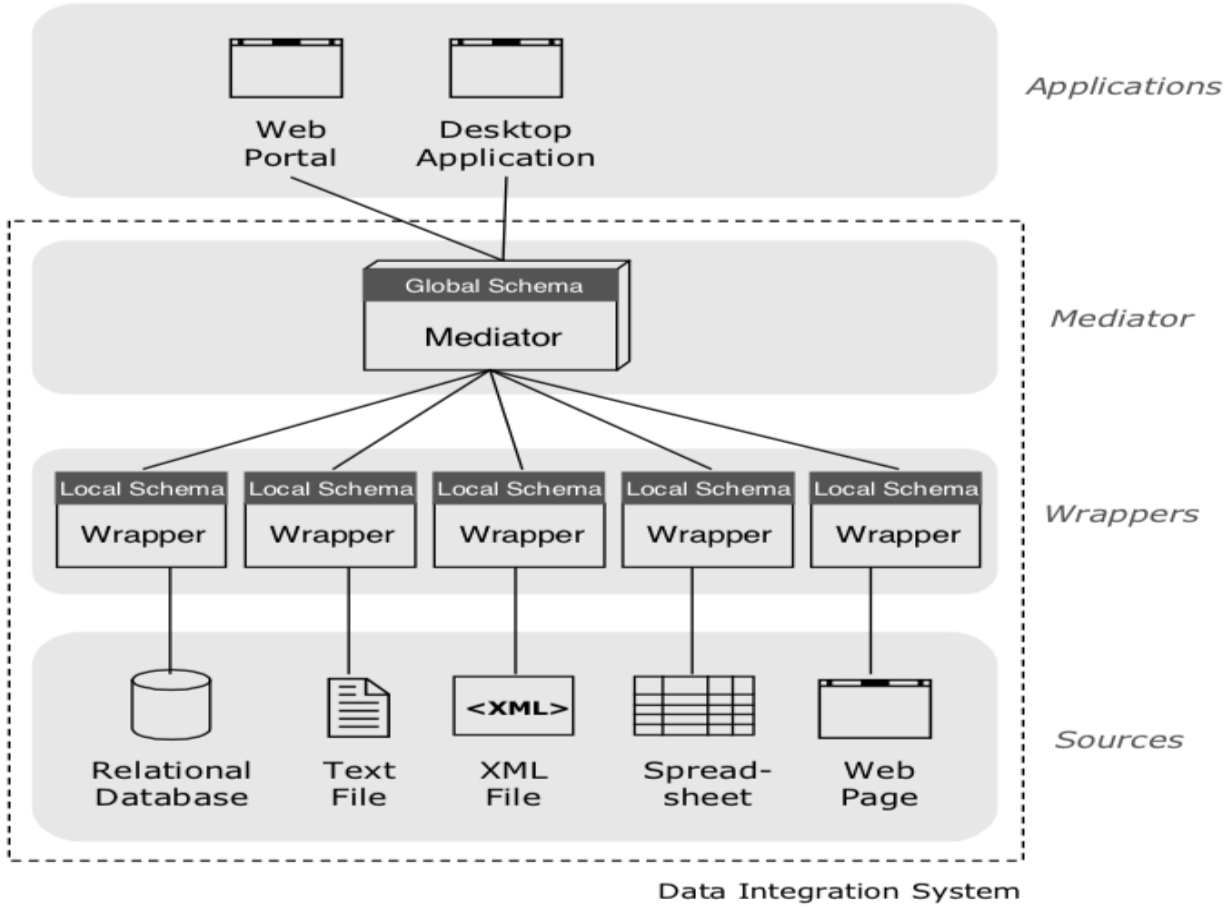
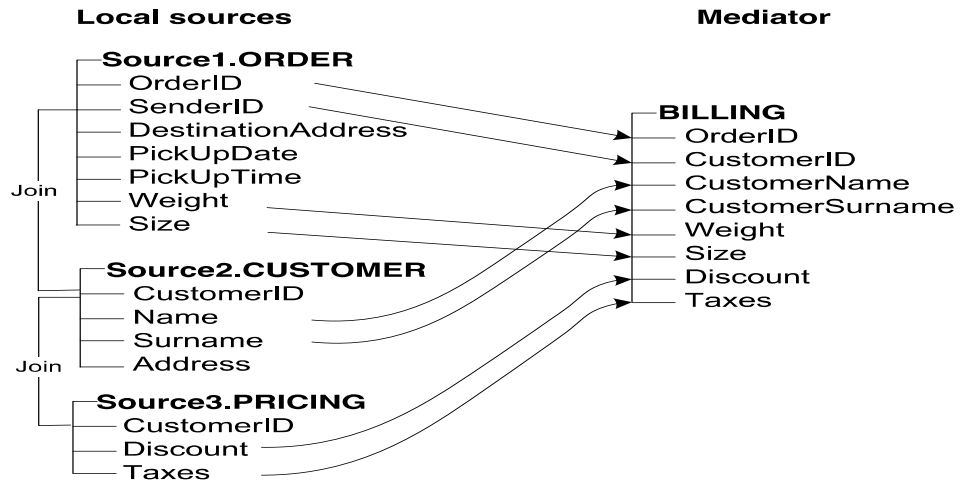


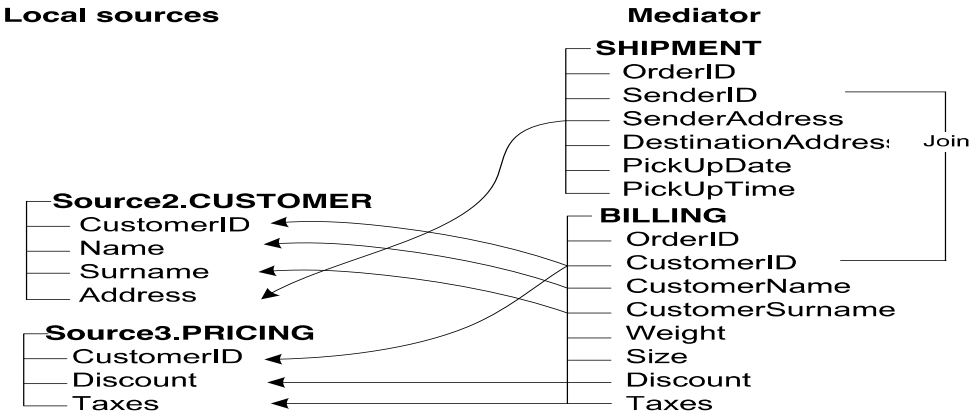
Figure 1: View-based Data Integration System (VDIS) Architecture

Picture from View-based Integration, Yannis Katsis, CSE, UC San Diego

Data Level Integration …



a) **GAV Mapping for the global relation BILLING**. The global relation is defined as a view on the local source relations.



b) **LAV Mapping for Source2 and Source3.** The local source relations are defined as views over the global relations.

Fig. 2.3 Example of GAV and LAV schema mappings for the integrated order DB.

Picture from Mashups: Concepts, Models and Architectures

Data Level Integration …

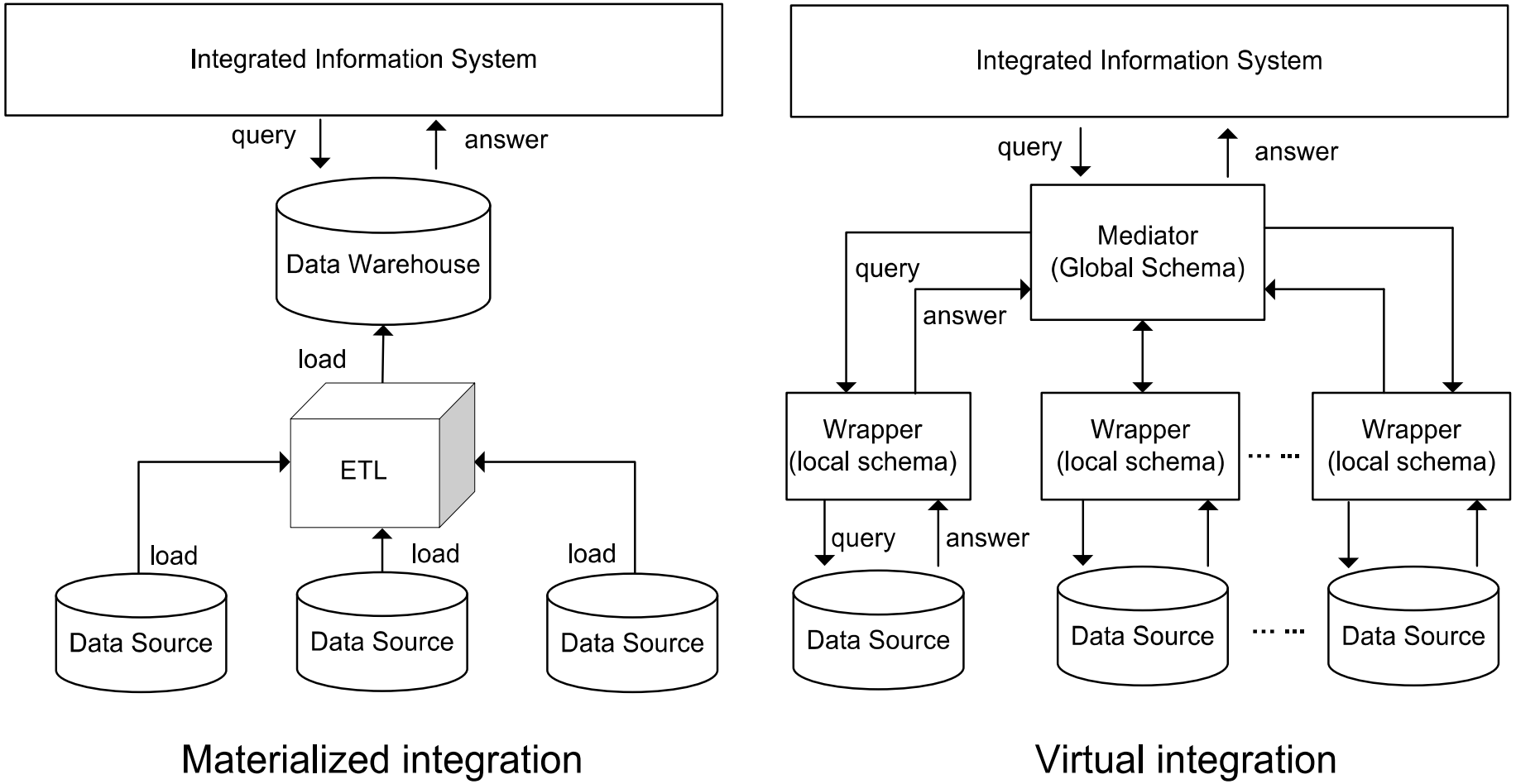


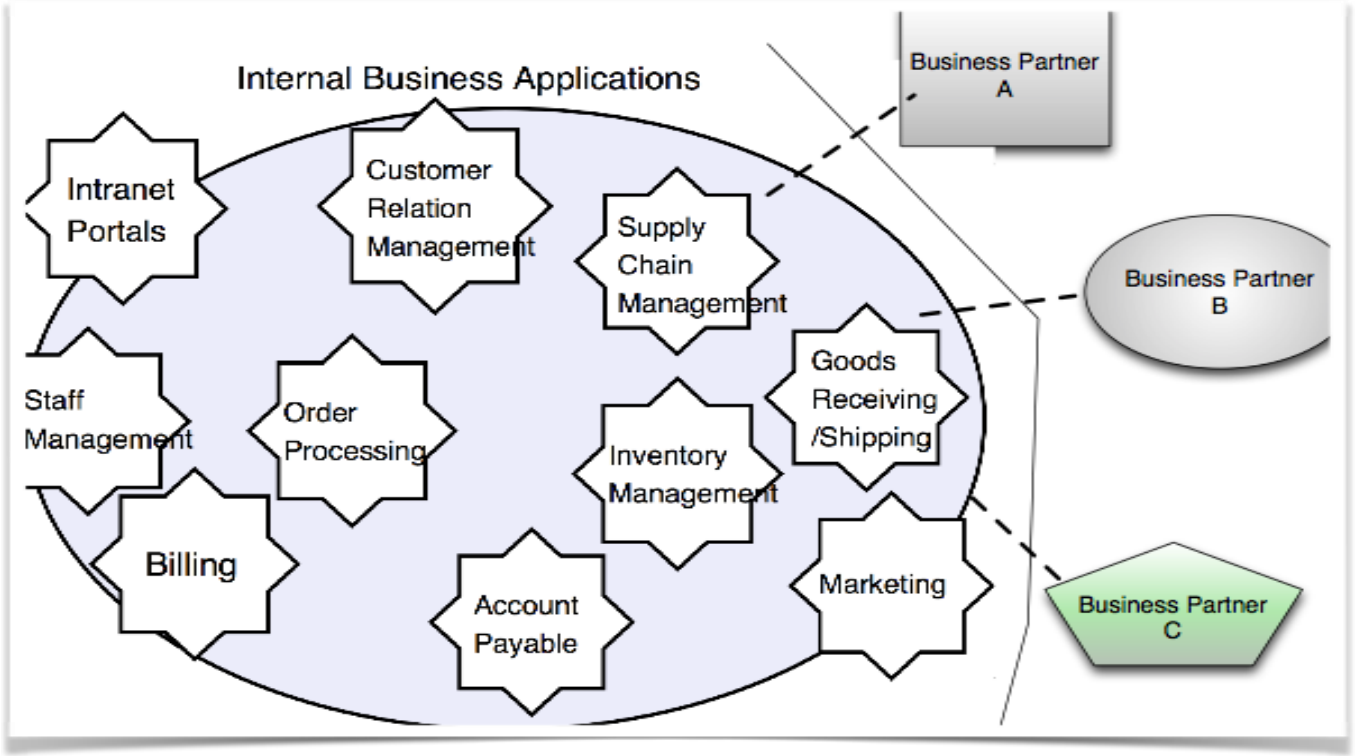
Fig. 2.1 Typical architectures for materialized and virtual data integration [3].

Picture from Mashups: Concepts, Models and Architectures

System Level Integration …

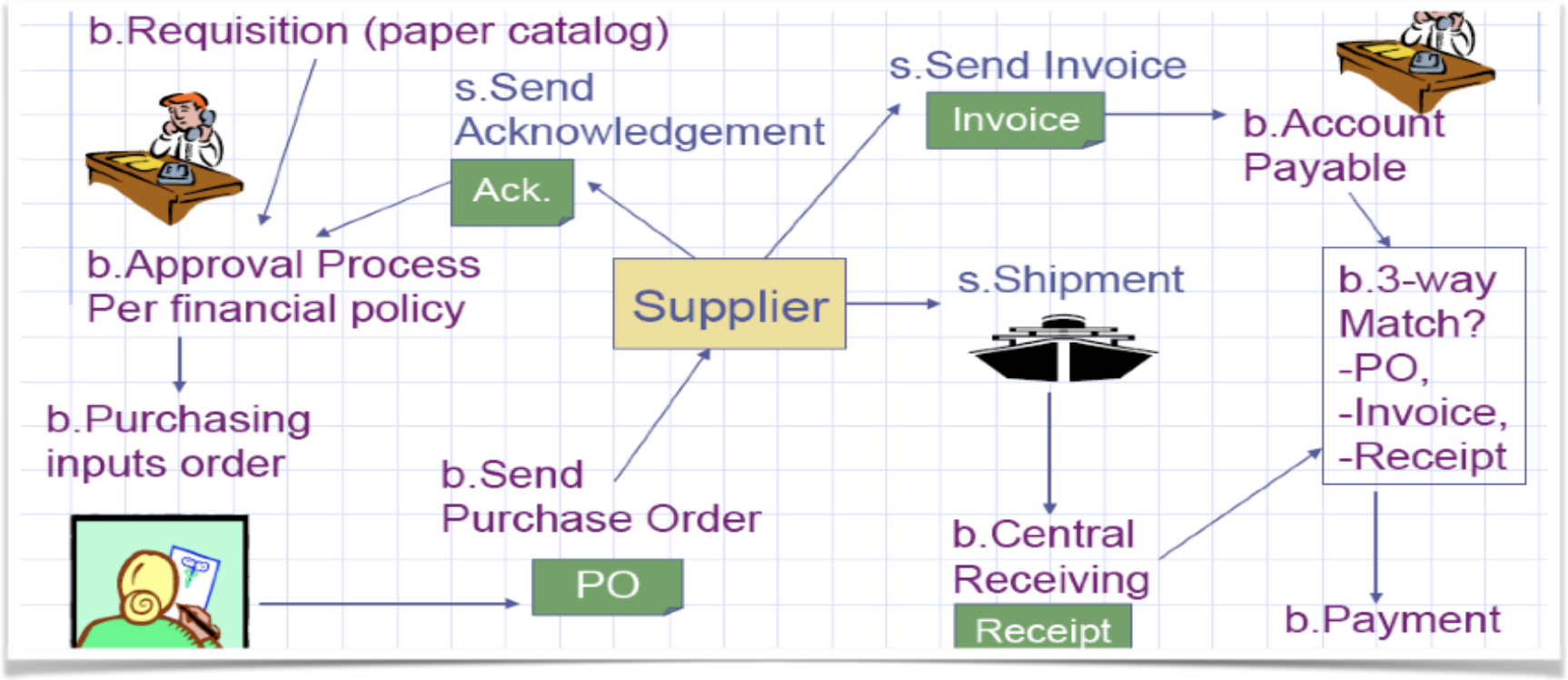
In enterprise environments, pick any sizeable organisation. You will see many departments performing different functionality

In silos, often supported by software systems

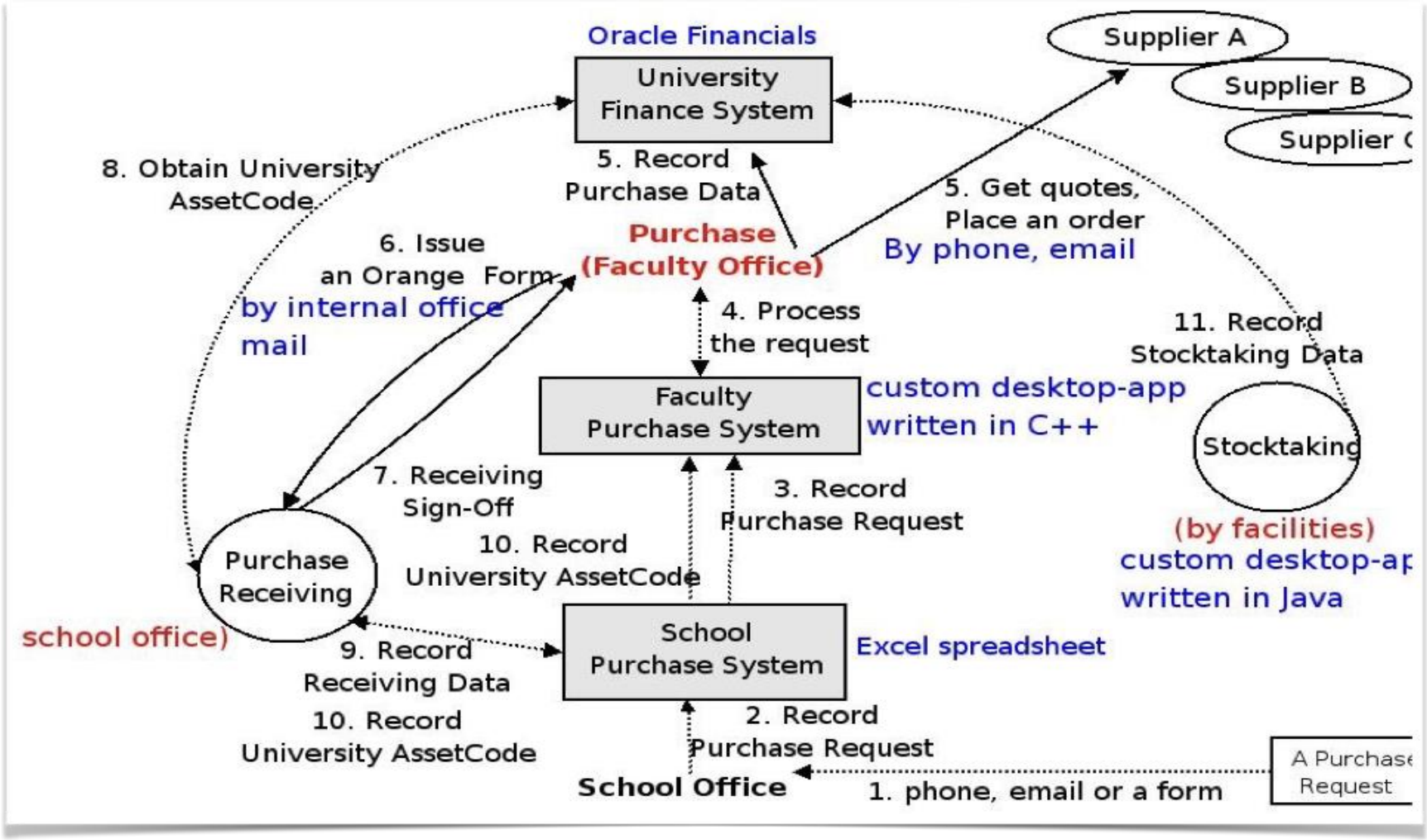


A Typical Purchase Order Process

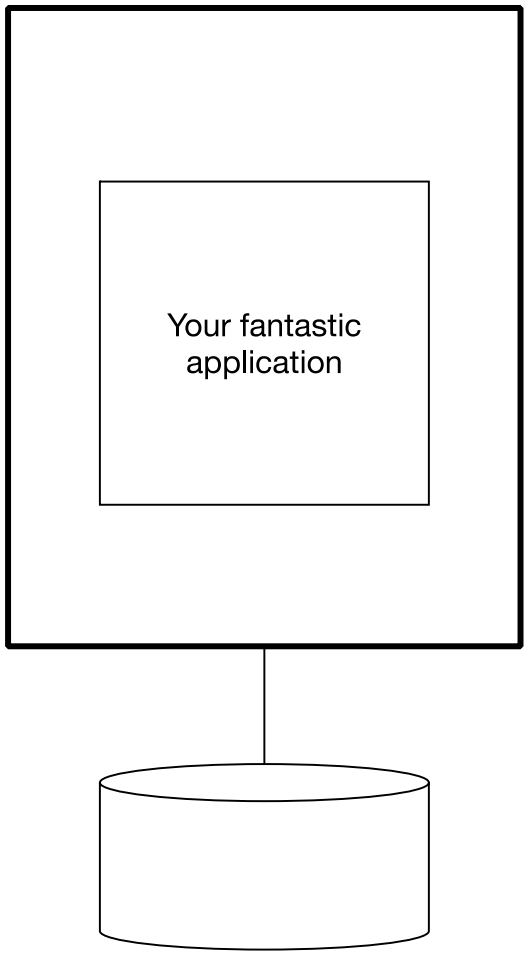
In reality: communication/coordination between the silos needed

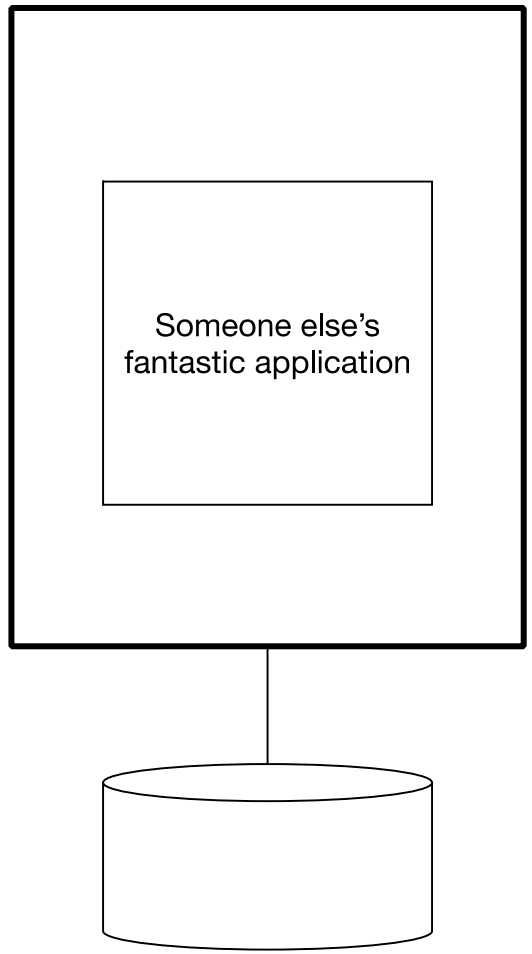


An example of (real) Purchase Order Process



Going outside of your system boundary …



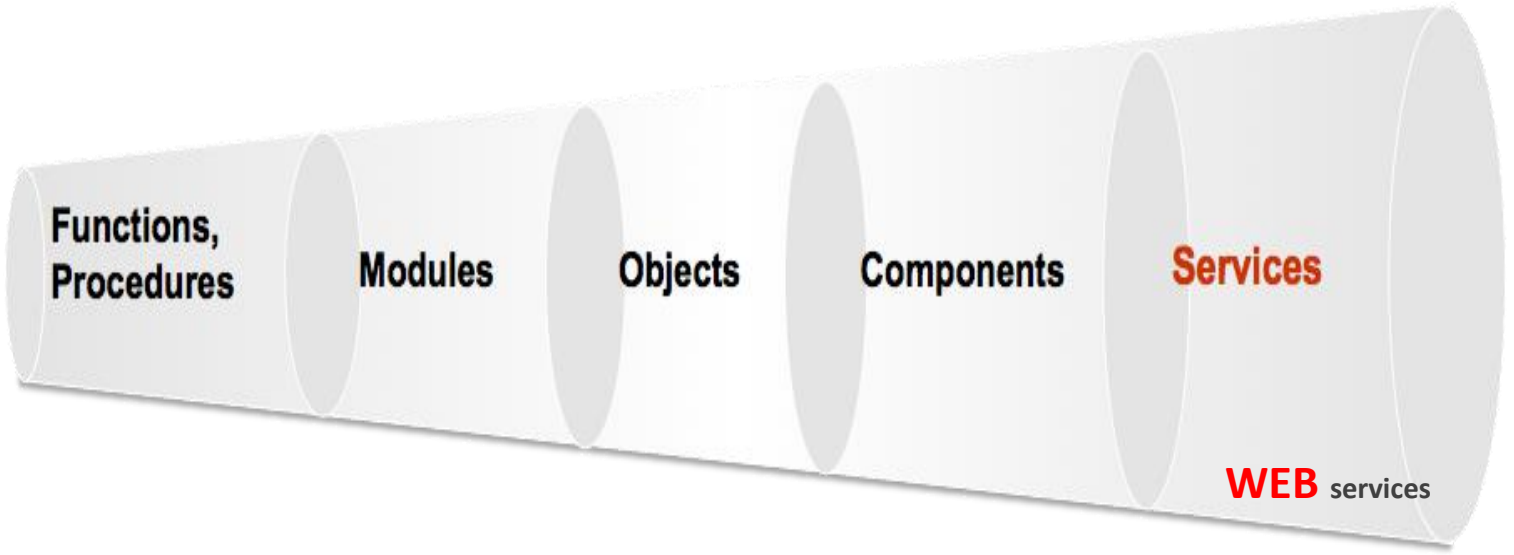


1. Same environment (language, platform)
2. Different environment
3. Messaging (how?)

The evolution of programming abstractions

Services: “customer” and “service provider”

Lines of code vs. Services - consider software building exercise as ‘building services’, ‘discovering services’ and ‘combining services’



Web = platform/language neutral

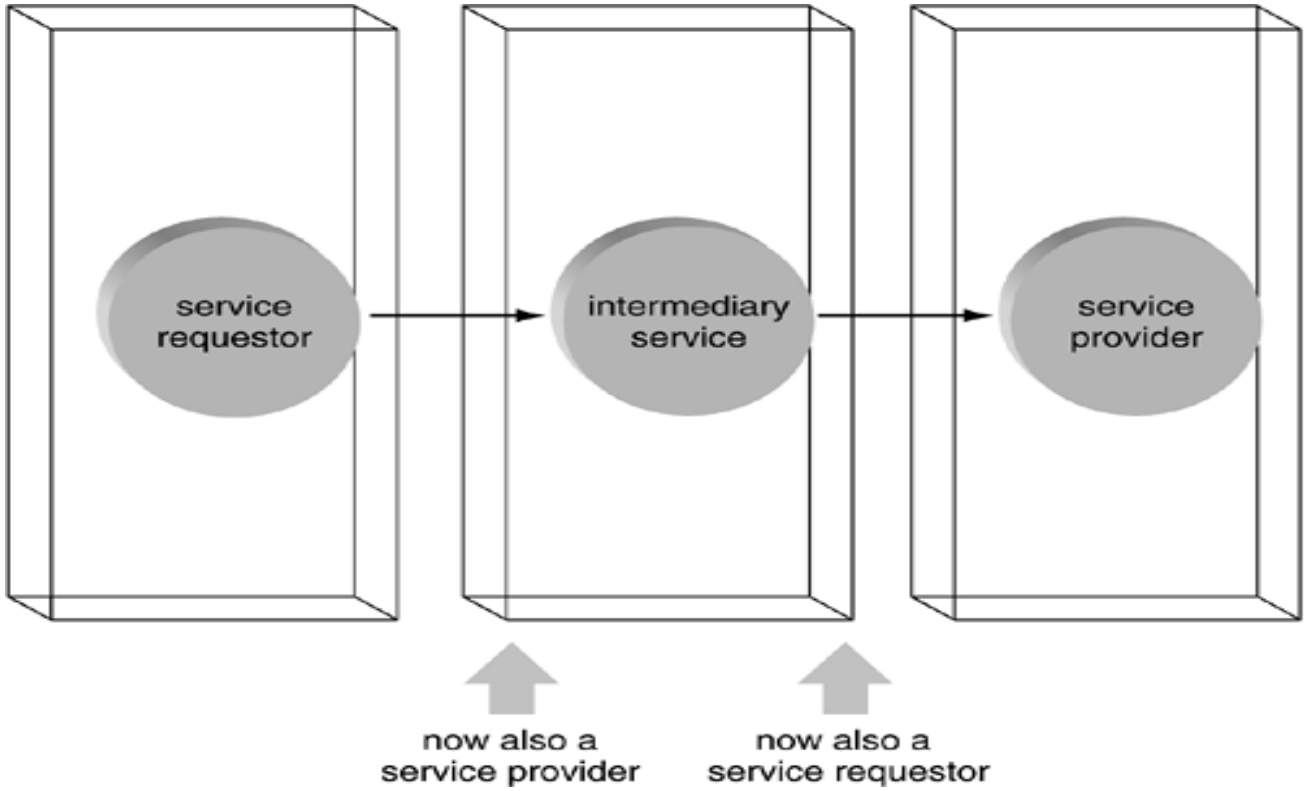
The evolution of programming abstractions

In SOA, we talk about software as a service ... That is, SOA is about building software systems composed of a collection of (software) services

A software service:

- A software asset that is deployed at an endpoint and is continuously maintained by a provider for user by one or multiple clients
- Services have explicit contracts that establish their purpose and how they should be used
- Software services are (supposed to be) reusable (“compose-able”) …
  - like lego blocks
  - “my” (the developer) service could be used in scenarios that I never anticipated

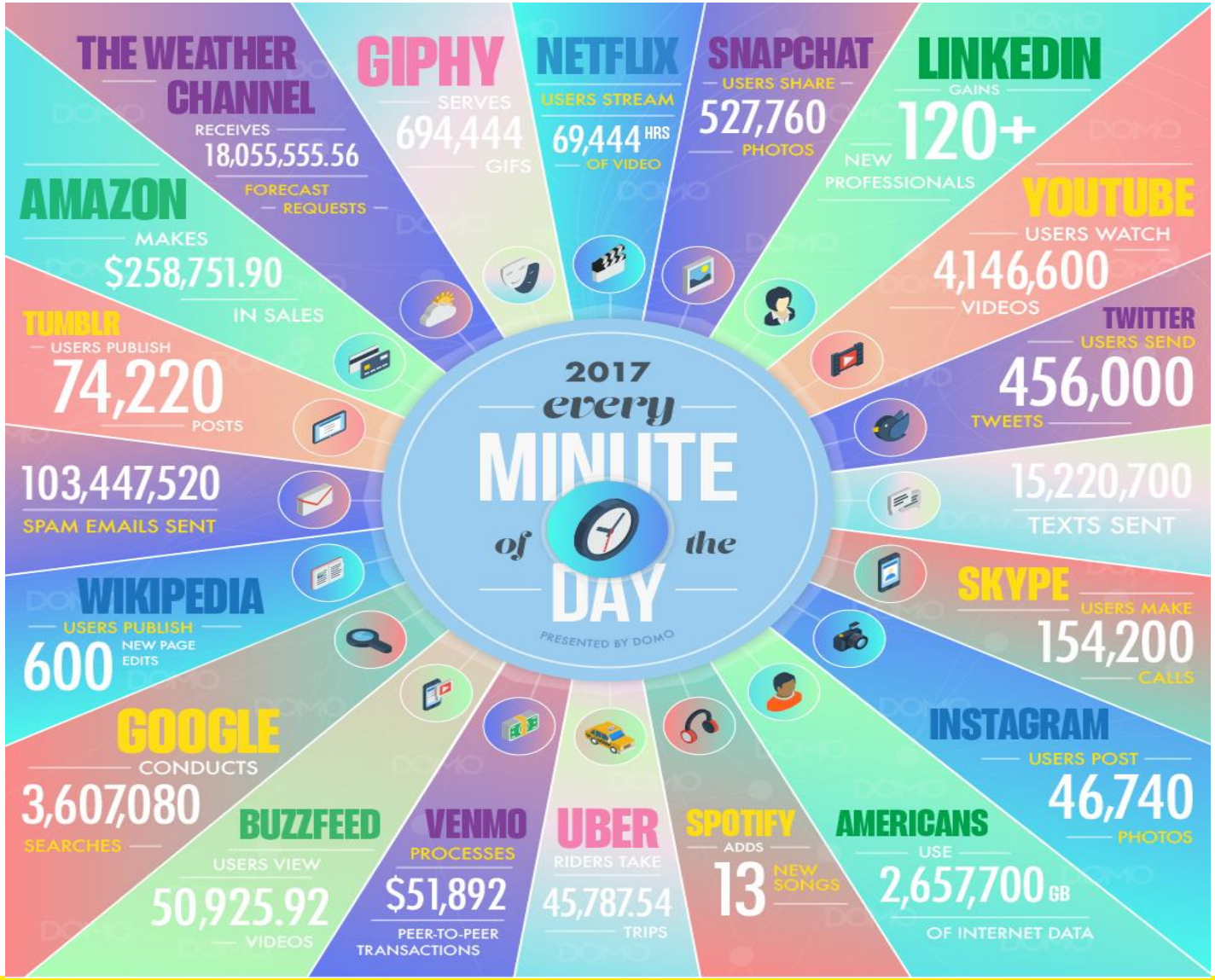
Simplified view of services (or API ?!)



Service-orientation - a way of integrating your applications as a set of linked services. If you can define the services, you can begin to link the services to realise more complicated 'services'

So again…Why Data Services?

Data Every Minute



Source: https://www.domo.com/learn/data-never-sleeps-5?aid=ogsm072517_1&sf100871281=1

Where is this Data

Data can be found in many places:

- on the Web, possibly via an API
- in documents in a file system
- in spreadsheets
- in videos
- etc. etc. etc.

and in a variety of formats (JSON, XML, HTML, TEXT…etc.)

So What is Next?

- In order to build a data service you need to know how to work with data
  - Accessing the data from multiple sources
  - Cleansing the data (e.g., removing corrupted or useless data)
  - Manipulating the data (e.g., merging, transformation, normalization)
  - Presenting the data (visualization)

Useful Reading

- **View-based Data Integration**, Yannis Katsis
- **Mashups: Concepts, Models and Architectures,** Daniel, Florian, Matera, Maristella

Q&A